(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENTAMT

(12) **Offenlegungsschrift**

(10) **DE 197 23 676 A 1**

(51) Int. Cl.[6]:
G 06 F 9/445
G 06 K 19/07
// G06F 17/60

(21) Aktenzeichen: 197 23 676.6
(22) Anmeldetag: 5. 6. 97
(43) Offenlegungstag: 27. 8. 98

DE 197 23 676 A 1

Mit Einverständnis des Anmelders offengelegte Anmeldung gemäß § 31 Abs. 2 Ziffer 1 PatG

(71) Anmelder:
Siemens AG, 80333 München, DE

(72) Erfinder:
Baldischweiler, Michael, 81669 München, DE;
Sedlak, Holger, 85658 Egmating, DE;
Pfab, Stefan, 82049 Großhesselohe, DE

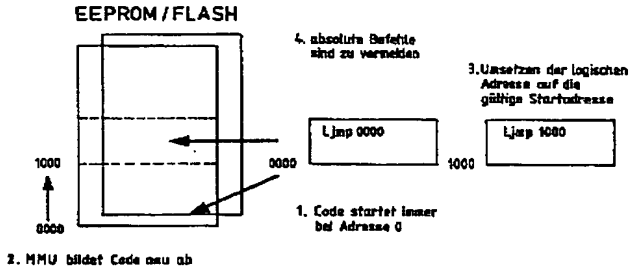
(56) Entgegenhaltungen:
DE 36 07 889 C2
DE 31 31 204 A1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) Verfahren zum Nachladen von Programmen auf eine Chip-Karte

(57) Die Erfindung betrifft ein Verfahren zum Nachladen von Programmen auf eine Chip-Karte, die eine CPU mit mehreren Adreßbereichen, insbesondere mit einem Programm- und einem Datenadreßraum, eine Speicherverwaltungseinheit und einen Adreß-Addierer aufweist, ein Betriebssystem zum Laden von Programmen an physikalischen Adressen seiner Wahl enthält, wobei das Verfahren gemäß einem ersten Aspekt der Erfindung die Schritte aufweist:
- Legen der nachladbaren Programme durch das Betriebssystem auf physikalische Start-Adressen,
- Anlegen einer Tabelle der physikalischen Start-Adressen und Verwalten dieser Adressen durch die Speicherverwaltungseinheit,
- Addieren der physikalischen Start-Adressen aus der Tabelle mit logischen Adressen in dem Adreß-Addierer zum Umsetzen in physikalische Adressen, und
- Ablegen der physikalischen Adressen in einem logischen Adreßraum.



DE 197 23 676 A 1

Beschreibung

Die Erfindung betrifft ein Verfahren zum Nachladen von Programmen auf eine Chip-Karte.

Neuere Entwicklungen auf dem Gebiet der Chip-Kartenanwendungen erfordern die Nachladbarkeit von Programmen auf der Chip-Karte im Feldeinsatz. Es ist daran gedacht, dem Kunden die Möglichkeit zu geben, Programme seiner Wahl auf die Chip-Karte zu laden. Dies ist bislang aus folgendem Grund nicht möglich: Jedes Programm basiert auf Adressen, an deren Position das Programm abgearbeitet wird. Ein sogenannter "Linker" legt diese Adressenzuordnung fest. Da für das nachzuladende Programm völlig unbekannt ist, welche Adressen in der Chip-Karte bereits belegt sind, muß die Möglichkeit geschaffen werden, nachzuladende Programme auf beliebigen Adressen ablaufen lassen zu können; d. h. die nachzuladenden Programme müssen auf der Chip-Karte relokatierbar sein.

Eine Aufgabe der vorliegenden Erfindung besteht demnach darin, ein Verfahren zu schaffen, durch das Programme auf Chip-Karte problemlos nachgeladen werden können.

Gelöst wird diese Aufgabe durch die Merkmale des Anspruchs 1 bzw. des Anspruchs 5.

Die Erfindung stellt demnach zwei Lösungsansätze für eine problemlose Relokatierbarkeit von nachladbaren Programmen auf einer Chip-Karte bereit, deren gemeinsame Idee darin besteht, die nachladbaren Programme durch das auf der Chip-Karte enthaltene Betriebssystem auf freie physikalische Adressen bzw. physikalische Segmente mit Hilfe von logischen Adressen bzw. logischen Segmenten zu verteilen. Der erste Aspekt der Erfindung ist Gegenstand des Anspruchs 1 und der zweite Aspekt der Erfindung ist Gegenstand des Anspruchs 5.

Weiterbildungen der Erfindung sind Gegenstand der Unteransprüche.

Nachfolgend wird die Erfindung anhand der nachfolgenden Figuren beispielhaft näher erläutert. Es zeigen:

Fig. 1 schematisch das Prinzip des erfindungsgemäßen Verfahrens gemäß dem ersten Aspekt,

Fig. 2 schematisch ein Ausführungsbeispiel des Verfahrens gemäß **Fig.** 1,

Fig. 3 schematisch das Prinzip des erfindungsgemäßen Verfahrens gemäß dem zweiten Aspekt,

Fig. 4 schematisch ein Ausführungsbeispiel des Verfahrens gemäß **Fig.** 3.

Gemäß dem ersten Aspekt der Erfindung ist demnach mit anderen Worten vorgesehen, daß der Programm-Code stets mit einer festen Adresse, in der Regel mit der Adresse 0 startet, wobei eine Hardware-Einheit die Programme auf verschiedene Programmbänke verteilt. Dieser Lösungsansatz wird im folgenden in bezug auf eine 8051-CPU erläutert.

Die 8051-CPU enthält einen Programm- und einen Datenadreßraum. Die nachfolgende Erläuterung des ersten erfindungsgemäßen Aspekts erfolgt anhand der 8051-CPU lediglich beispielhaft aus Gründen der Verdeutlichung der Erfindung; d. h. der erste Aspekt der Erfindung ist ohne weitere auf andere Adreßbereiche grundsätzlich übertragbar. Neben dem Programm-Code gilt der Vorschlag zur Relokatierbarkeit gemäß dem ersten Aspekt der Erfindung (und auch gemäß dem zweiten Aspekt der Erfindung) auch für Datenbereiche.

Gemäß dem ersten Aspekt der Erfindung (vgl. hierzu die **Fig.** I und 2) werden also zunächst sämtliche nachladbaren Programme auf eine feste Start-Adresse, beispielsweise auf die Start-Adresse 0000, gelegt (Schritt 1). Das On-Chip-Betriebssystem lädt daraufhin die Programme an physikalische Startadressen (> 64k möglich) seiner Wahl. In der Speicherverwaltungseinheit bzw. Memory Management Unit wird daraufhin eine Tabelle angelegt, welche die physikalischen Startadressen dieser Programme verwaltet (Schritt 2).

Mit einem Addierer werden daraufhin die logischen Adressen (maximal 64k) der Programme (beginnend bei 0) auf die jeweils gültige Startadresse umgesetzt gelegt (Schritt 3). Dabei wird in dem logischen 64k-8051 Adressenraum genau ein Programm abgelegt. Durch die Auswahl einer anderen physikalischen Startadresse wird ein anderes Programm im 64k-Adreßraum abgelegt. Hier ist darauf zu achten, daß keine absoluten Befehle verwendet werden.

Der physikalische Speicher muß nicht notwendigerweise in 64k Programmbänke organisiert sein. Mit einer geeigneten Hardware-Einheit, die eine Programmlängenüberwachung vornimmt, können die physikalischen Programme vielmehr vorteilhafterweise an die tatsächliche Programmgröße angepaßt werden.

Im folgenden wird ebenfalls unter bezug auf die 8051-CPU der zweite Aspekt der vorliegenden Erfindung gemäß Anspruch 5 beispielhaft erläutert, der kurz gesagt darin besteht, daß eine Hardware-Einheit die Adreßlage der Programme innerhalb des logischen 64K-Adreßraums verändert.

Zunächst werden gemäß **Fig.** 3 und 4 sämtliche nachladbaren Programme auf beliebige Segmentgrenzen gelegt (beispielsweise 1k, 2k, 3k,. . .). Das On-Chip-Betriebssystem verteilt daraufhin die Programme auf die jeweils noch freien Segmente. In einer Tabelle erfolgt dann die Zuordnung von dem logischen Segment, auf das gelinkt wurde, zu dem physikalischen Segment, welches für das Programm noch frei war. Bei dieser Variante entfällt das Addieren von physikalischen Start-Adressen wie gemäß dem ersten Aspekt der vorliegenden Erfindung. Statt dessen werden die höchstwertigen, beispielsweise die beiden höchstwertigen Adreß-Bits geeignet modifiziert. Im in **Fig.** 3 und 4 gezeigten Beispiel sind vier Segmente dargestellt. Die notwendige Modifikation der beiden höchstwertigen Bits wird in einer Tabelle abgelegt.

Beim Nachladen des Programms erkennt das On-Chip-Betriebssystem dessen logische Adreßlage und außerdem die noch freien Segmente.

Bei den in **Fig.** 3 und 4 gezeigten Beispielen wird das physikalische Segment 0 mit der logischen Adresse 1 belegt. Deshalb müssen bei der Programmausführung dieses Segments die beiden höchstwertigen Bits der 8051-CPU von 01 auf 00 modifiziert werden. Obwohl das Programm logisch auf 01xxxxxx abläuft, liegt es physikalisch auf 00xxxxxx.

Beiden Aspekten der vorliegenden Erfindung ist gemeinsam, daß die Zuordnung von Programmen zu Adressen ausschließlich durch Befehle erfolgt, die absolute Adressen benötigen, hier z. B. Ljmp adr 16, Lcal adr 16. Das heißt, absolute Befehle werden zur Zuordnung der Programme zu Adressen vorteilhafterweise vermieden.

Wenn diese Befehle durch PC-relative Befehle ersetzt werden, können die Programme auch an beliebige Stellen verschoben werden. Die Adressierung wird immer relativ zum PC vorgenommen und ist damit unabhängig von der absoluten Adreßlage: Rjmp rel 16, Rcal rel 16.

Für die Erfindung gemäß beiden Aspekten gilt demnach, daß das Betriebssystem nur den Startpunkt der nachladbaren Programme kennen muß.

Patentansprüche

1. Verfahren zum Nachladen von Programmen auf eine Chip-Karte, die eine CPU mit mehreren Adreßbereichen, insbesondere mit einem Programm- und einem Datenadreßraum, eine Speicherverwaltungseinheit und einen Adreß-Addierer aufweist, ein Betriebssystem

zum Laden von Programmen an physikalischen Adressen seiner Wahl enthält, und folgende Schritte umfaßt:

- Legen der nachladbaren Programme durch das Betriebssystem auf physikalische Start-Adressen,
- Anlegen einer Tabelle der physikalischen Start-Adressen und Verwalten dieser Adressen durch die Speicherverwaltungseinheit,
- Addieren der physikalischen Start-Adressen aus der Tabelle mit logischen Adressen in dem Adreß-Addierer zum Umsetzen in physikalische Adressen, und
- Ablegen der physikalischen Adressen in einem logischen Adreßraum.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß der logische Adreßraum durch einen physikalischen Speicher auf der Chip-Karte festgelegt ist.

3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß der physikalische Speicher in Programmbänke organisiert ist.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß eine Hardware-Einheit zur Programmlängen-Überwachung vorgesehen ist, um die physikalischen Programmbänke an die tatsächliche Programmgröße anzupassen.

5. Verfahren zum Nachladen von Programmen auf eine Chip-Karte, die eine CPU mit mehreren Adreßbereichen, insbesondere mit einem Programm- und einem Datenadreßraum, einen Speicher, und eine Speicherverwaltungseinheit aufweist, ein Betriebssystem zum Laden von Programmen an physikalischen Adressen seiner Wahl enthält, und folgende Schritte umfaßt:

- Legen bzw. Linken der nachladbaren Programme auf beliebige Segmentgrenzen bzw. logische Segmente des Speichers,
- Verteilen der nachladbaren Programme durch das Betriebssystem auf die jeweils noch freien physikalischen Segmente,
- Zuordnen der logischen Segmente auf die physikalischen Segmente in einer Tabelle unter Modifizieren der höchstwertigen Adreß-Bits durch die Speicherverwaltungseinheit.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, daß die nachgeladenen Programme durch eine Rücknahme der Adreß-Bit-Modifikation ausgeführt werden.

7. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, daß die nachgeladenen Programme den Adressen bzw. Segmenten ausschließlich durch absolute Adressen benötigende Befehle (d. h. unter Vermeidung absoluter Befehle) zugeordnet werden.

8. Verfahren nach einem der Ansprüche 1 bis 6, dadurch gekennzeichnet, daß die Programmzuordnung jeweils bei Ziffer 0 beginnt.

Hierzu 4 Seite(n) Zeichnungen

Nummer: DE 197 23 676 A1
Int. Cl.[6]: G 06 F 9/445
Offenlegungstag: 27. August 1998

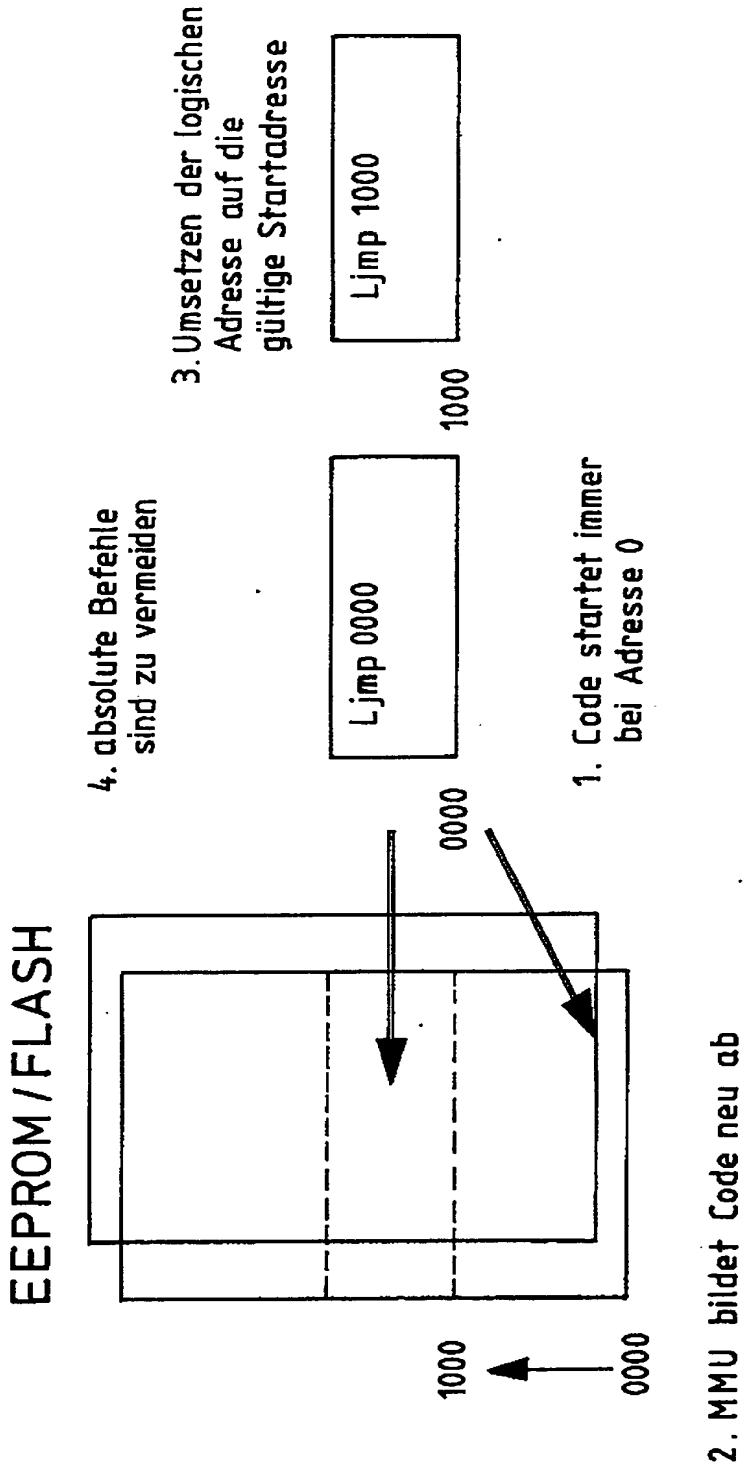


Fig. 1

Nummer: DE 197 23 676 A1
Int. Cl.[6]: G 06 F 9/445
Offenlegungstag: 27. August 1998

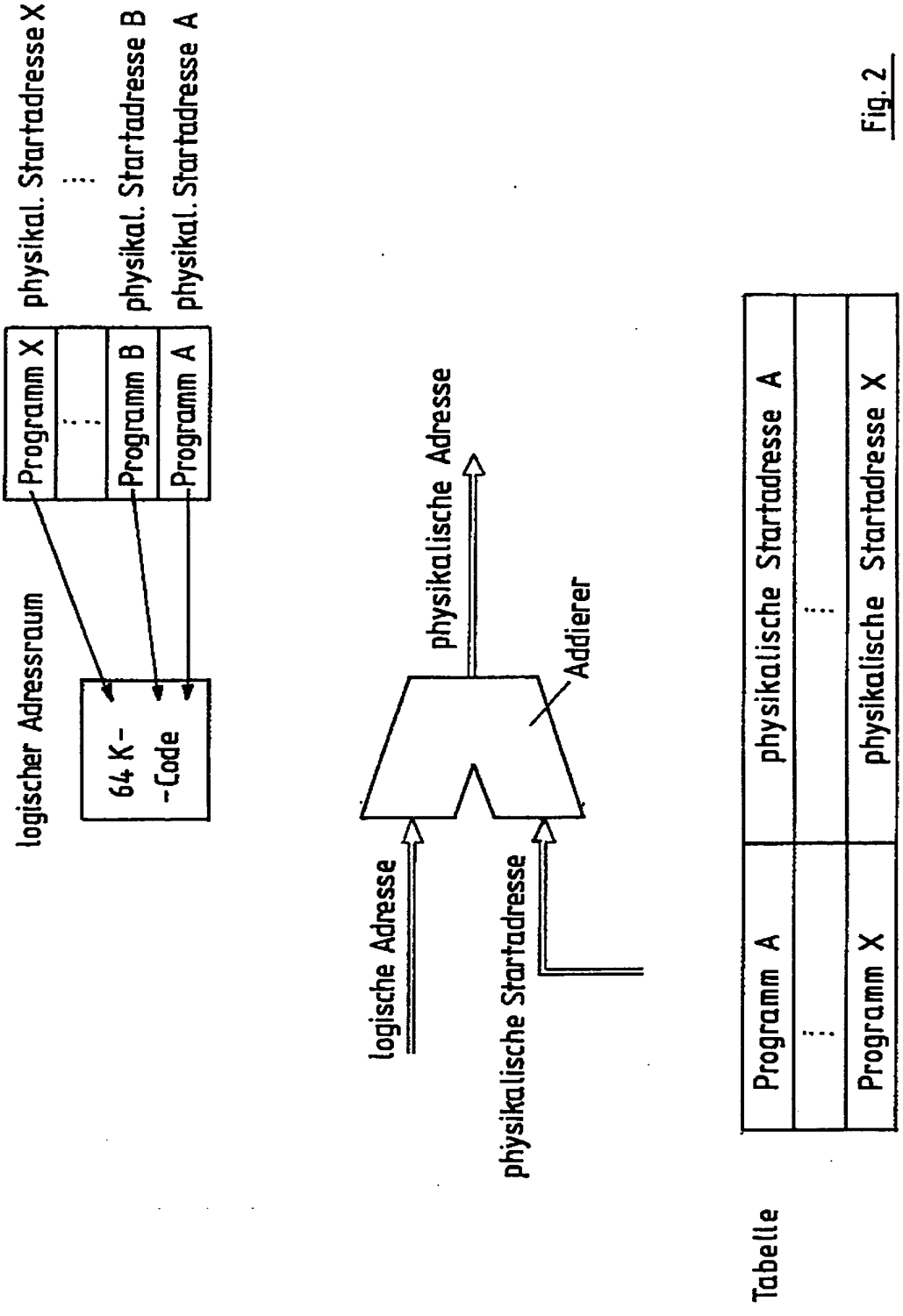


Fig. 2

Nummer: DE 197 23 676 A1
Int. Cl.[6]: G 06 F 9/445
Offenlegungstag: 27. August 1998

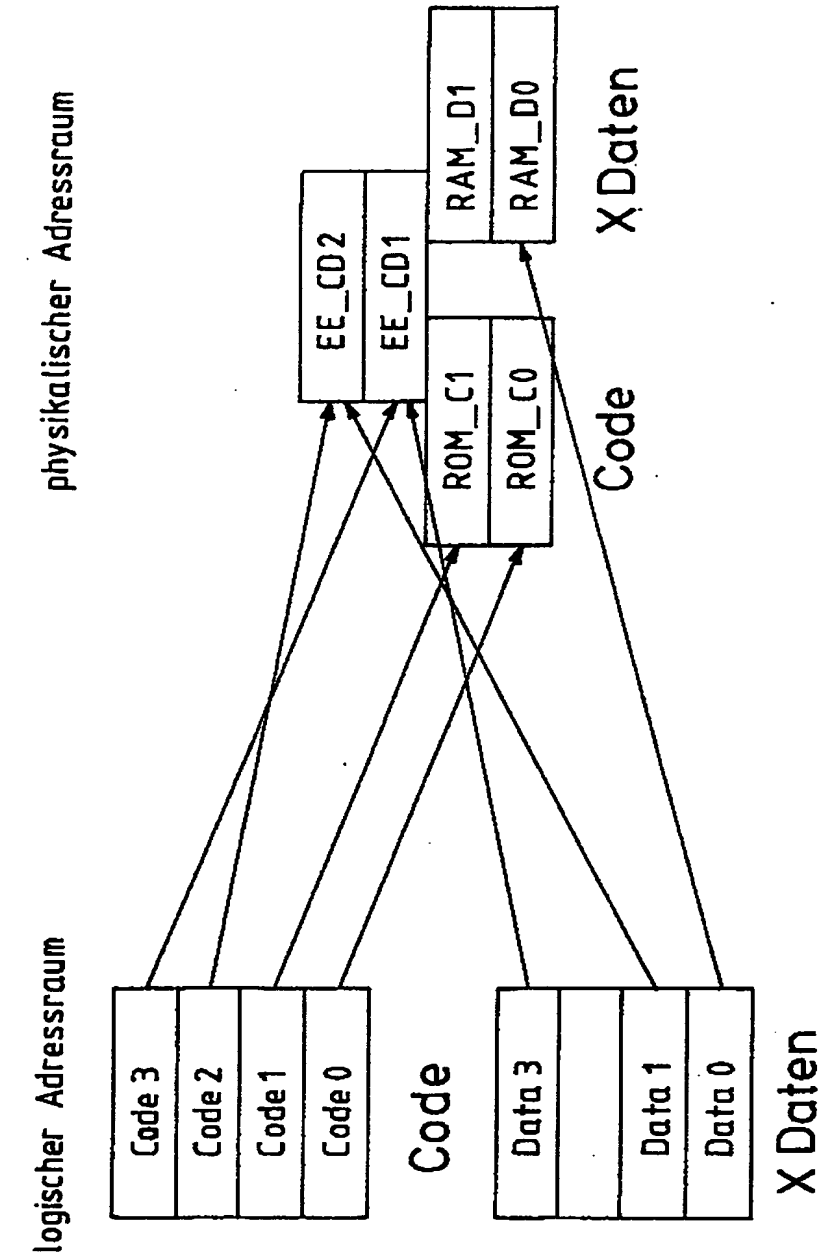


Fig. 3

Segmentierung des 8051 Adressraums für Multifunktionale Chipkarten

Nummer: DE 197 23 676 A1
Int. Cl.[6]: G 06 F 9/445
Offenlegungstag: 27. August 1998

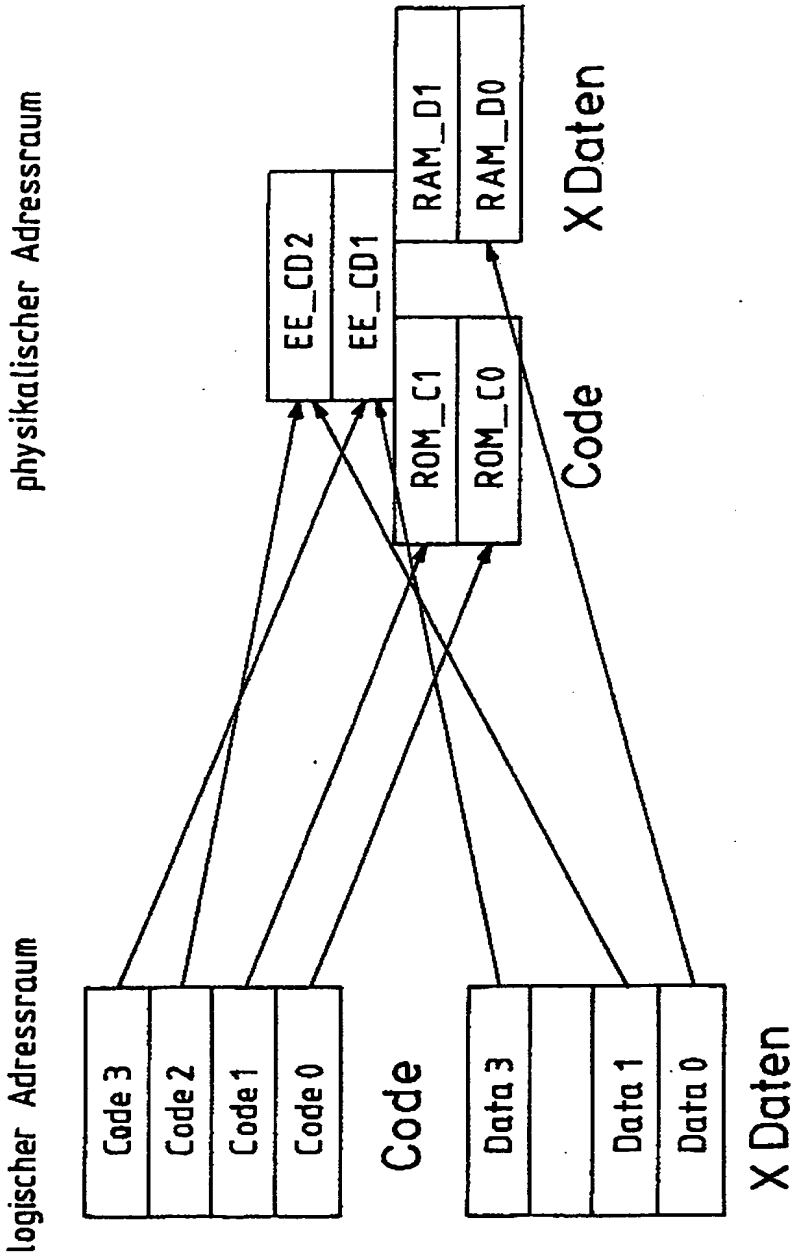


Fig. 3

Segmentierung des 8051 Adressraums
für Multifunktionale Chipkarten